Хранилища

Key Value

> Per.10: Rely on the static type system

Ожидание?

```
INSERT INTO user (id, name) VALUES (1, 'c++');
SELECT * FROM user WHERE id = 1;
```

Курсор, причем тут игровые приставки?

```
SELECT * FROM user;
```

ALTER TABLE

course

id	name
1	c++
2	python

```
char *raw =
"00001c++                                                                "
"00002python                                                             "
;
```

course

id	name	level
1	c++	advanced
2	Python	novice

```
char *raw =
"00001c++                                                               advanced          "
"00002python                                                            novice            "
;
```

```
"00001c++                                                                "
"00001advanced         c++                                                                "
```

course (2)

id	name	level_id
1	c++	1
2	python	2

level | id | level | | —- | ——– | | 1 | advanced | | 2 | novice |

course (3)

id	name
1	c++, advanced
2	python, novice

course

id	Value
1	{“name”:“c++”,“level”:“adv”}
2	Python

group

id	name	course_id
1	2017-12	1
2	2018-02	1
3	2016-12	2
4	2017-12	2

value

Table	field	id	Value
Course	Level	1	Advance
Course	Level	2	Novice
Group	Visible	2	Off
Course	Name	1	c++
Course	Name	2	Python

SELECT * FROM course WHERE id=1;

SELECT * FROM value WHERE table=‘Course’ AND id=1;

MongoDB / Create a Document

```
auto builder = bsoncxx::builder::stream::document{};
bsoncxx::document::value doc_value = builder
  << "name" << "MongoDB"
  << "type" << "database"
  << "count" << 1
  << "versions" << bsoncxx::builder::stream::open_array
    << "v3.2" << "v3.0" << "v2.6"
  << close_array
  << "info" << bsoncxx::builder::stream::open_document
    << "x" << 203
    << "y" << 102
  << bsoncxx::builder::stream::close_document
  << bsoncxx::builder::stream::finalize;

bsoncxx::stdx::optional<mongocxx::result::insert_one> result =
 restaurants.insert_one(doc_value);
```

MongoDB / Query

```
mongocxx::cursor cursor = collection.find(
  document{} << "count" << open_document <<
    "$gt" << 50 <<
    "$lte" << 100
  << close_document << finalize);
for(auto doc : cursor) {
  std::cout << bsoncxx::to_json(doc) << "\n";
}
```

Redis / Commads

- Geo

- Hashes
- HyperLogLog
- Lists
- Pub/Sub
- Scripting
- Sets
- Sorted Sets
- Strings
- Transactions

Redis / Simple pipeline

```
redisReply *reply;
redisAppendCommand(context,"SET foo bar");
redisAppendCommand(context,"GET foo");
redisGetReply(context,&reply); // reply for SET
freeReplyObject(reply);
redisGetReply(context,&reply); // reply for GET
freeReplyObject(reply);
```

Aerospike / C client

```
as_config_init(&config);

config.hosts[0] = { .addr = "127.0.0.1", .port = 3000 };
aerospike_init(&as, &config);
aerospike_connect(&as, &err);

as_key_init(&key, "test", "demo_set", "test_key");

as_record_inita(&rec, 2);
as_record_set_int64(&rec, "test-bin-1", 1234);
as_record_set_str(&rec, "test-bin-2", "test-bin-2-data");

aerospike_key_put(&as, &err, NULL, &key, &rec);

aerospike_close(&as);
aerospike_destroy(&as);
```

LevelDB / Atomic Updates

```
#include <leveldb/write_batch.h>

std::string value;
leveldb::Status s = db->Get(leveldb::ReadOptions(), key1, &value);
if (s.ok()) {
  leveldb::WriteBatch batch;
  batch.Delete(key1);
  batch.Put(key2, value);
  s = db->Write(leveldb::WriteOptions(), &batch);
}
```

Какие знаем еще?

- Колоночные, аналитические
- Временные ряды
- Обратные индексы, поисковые
- Service Discovery
- Configuration
- GIT
- Как выбрать-то?

Полистать

- https://mongodb.github.io/mongo-cxx-driver/mongocxx-v3/tutorial/
- https://www.aerospike.com/docs/client/c/index.html
- https://github.com/redis/hiredis
- https://github.com/google/leveldb/blob/master/doc/index.md

Спасибо за внимание!